Stop-words in Keyphrase Extraction Problem

S. Popova[1,2], L. Kovriguina[2], D. Mouromtsev[1]
[1]Saint Petersburg National Research University of Information Technologies, Mechanics and Optics,
[2]Saint-Petersburg State University
Saint-Petersburg, Russia
svp@list.ru, {lkovriguina, d.muromtsev}@gmail.com

I.Khodyrev
Saint-Petersburg State Electrotechnical University,
VISmart
Saint-Petersburg, Russia
kivan.mih@gmail.com

***Abstract*—Keyword extraction problem is one of the most significant tasks in information retrieval. High-quality keyword extraction sufficiently influences the progress in the following subtasks of information retrieval: classification and clustering, data mining, knowledge extraction and representation, etc. The research environment has specified a layout for keyphrase extraction. However, some of the possible decisions remain uninvolved in the paradigm. In the paper the authors observe the scope of interdisciplinary methods applicable to automatic stop list feeding. The chosen method belongs to the class of experiential models. The research procedure based on this method allows to improve the quality of keyphrase extraction on the stage of candidate keyphrase building. Several ways to automatic feeding of the stop lists are proposed in the paper as well. One of them is based on provisions of lexical statistics and the results of its application to the discussed task point out the non-gaussian nature of text corpora. The second way based on usage of the Inspec train collection to the feeding of stop lists improves the quality considerably.**

Keywords—keyphrase extraction, keyphrase identification, stop words extraction, informational retrieval, natural language processing.

I. INTRODUCTION

The key problem of this paper is keyphrase extraction for the abstracts of scientific publications. Automatic keyphrase extraction obtained, is desirable for subtasks of information retrieval: classification and clustering [1-3], data mining and knowledge extraction and representation, text summarization, data indexing [4]. Despite numerous researches and efforts, the quality of automatic keyphrase extraction is still far from being high.

The domain research environment has specified a layout for keyphrase extraction task. The task is usually divided into two parts: 1) extraction of candidate keyphrases; 2) classification of the extracted candidate set into keyphrases and non-keyphrases [5, 6] or ranking candidate keyphrases for further selection of n-best of them as keyphrases [7-11].

There is an alternative method based on ranking of the words of a document with further merging of selected words into keyphrases [12-14]. The third method dominantly used in search results clustering is based on Suffix Tree [1-3].

In previous papers [15-16] we used a simple algorithm to build a set of candidate keyphrases to annotate abstracts of scientific publications and obtained good results. The hold experiments show a good intersection between automatically selected non one-word sequences of nouns and adjectives and keyphrases assigned to the abstracts manually by the experts. However, there is "noise" among these candidate keyphrases. It consists of words of current usage like "experimental results", "good performance", etc. But it is clear, that an expert assigning keyphrases to the document will never mark such phrases as keyphrases irrespectively to the domain of the article. Thus, it is impossible to imagine that a human will identify the document with keyphrases "result of the experiment" or "previous research", because these collocations belong to the words of common usage and do not reveal the topic or domain of the document itself. In the present paper we consider the possibilities provided by the replacement of the stage of candidate phrase ranking by the stage of identifying the phrases that not at any price shall be added to the document as keyphrases. Such phrases shall not be built on the stage of building candidate keyphrases. The speculation of our research is: "whether excluding of phrases of common usage leads to quality improvement and whether it is possible to regard the rest of candidate keyphrases as real keyphrases?" Thus, the main task we concentrate in this paper is the challenge how to retrieve the set of candidate keyphrases and the documents so that no false keyphrases were left in them?

II. STATE-OF-THE-ART

There are two main trends in the keyword extraction domain. The first trend treats keyword extraction as a subtask, a necessary application to the main task of queries clustering or subtopics construction [1-3]. Existing algorithms have snippets to the queries in Google or Yahoo at the input and produce the most frequent word subsequences that are considered cluster labels. The procedure used for this task is Suffix Tree.

The second trend considers keyword extraction it's main task. In numerous papers words' ranking and weighing was proposed in order to select the words with the highest weights and merge those of the words in a phrase which

follow each other in the text. In [12] it has been proposed to build a text graph with whose vertices are words and weight each vertices with TextRank, that is the adapted PageRank. The paper [13] continues this approach taking the neighboring documents except the text itself to build the graph. The paper [14] uses both words and collocations as vertices and compares results given by TextRank and TF-IDF [17].

The dominant position is occupied by the approach that divides keyphrase extraction into two subtasks: building the set of candidate phrases and ranking the obtained set in order to select keyphrases. Formally, candidate phrases are either n-grams, or specified sequences, or both. Despite the way of building candidate phrases it has been shown that keyword phrases are mostly nouns and adjectives [6][11-14]. POS-tagging itself is an independent task requiring professional competence that is why the development of algorithms that need no POS-tagging is still actual [8]. Among the procedures used to filter out good keyphrases from the candidate set are Naïve Bayes model [5], part-of-speech information [6], genetic algorithms [18] e.t.c. The variety of ranking algorithms appeared in the domain is impressive. The procedures are mainly based on the information about the phrase length, its' first occurrence in the text from the beginning of the document, information about stop words in the phrase, IDF information, various text statistics and its' combinations, exterior data (e.g. Wikipedia) [10].

We have done several experiments in the paper [16] to compare ranking procedures based on the information about word frequencies to rank candidate phrases. Any sequence of adjectives and nouns having no delimiters (punctuation marks, stop words, other grammar classes) was considered a candidate keyphrase. The result was interesting and unobvious: all ranking procedures differ each other only in the way of ranking unigrams. If the latest are removed, the other phrases are ranked similarly. The experiments have also shown that unigram removal improves the quality of keyword extraction. It can be explained by the small usage of unigrams as keyphrases and large number of them in candidate set. That's the stimulus to use algorithms that consider the keyphrase length making the weight of unigrams lower. On the whole, the experiments have not pointed out to some remarkable ranking procedure so we decided to decline ranking and try to improve the quality of annotating by other methods. We held the working hypothesis that moving of phrases of common scientific usage to the stop list will improve the quality. Such phrases can be removed on the stage of candidate phrase building.

III. Evaluation

To evaluate the quality of candidate keyphrases we have chosen one of the most frequently used measures of quality estimating – F-score [17], that is a combination of two characteristics that is Precision and Recall of the automatically extracted keyphrases in respect to the manually extracted keyphrases:

$$\text{Precision} = (C \cap G)/G, \quad \text{Recall} = (C \cap G)/C,$$

$$\text{F-score} = (2*\text{Precision}*\text{Recall})/(\text{Precision}+\text{Recall})$$

where $C \cap G$ is the number of "true positives"– keyphrases that have been extracted properly in all considered documents, C — total number of manually extracted keyphrases in all considered documents, G – total number of automatically extracted keyphrases in all considered documents [6].

IV. DataSet

Inspec dataset – one of the popular collections used in researches for keyword extraction [6, 10, 12, 14]. It contains abstracts of journal scientific papers written in English. Abstracts are dated 1998-2002 and cover "Computers and Control" and "Information Technology" domains. Inspec dataset consists of three subcollections: training dataset (1000 abstracts), evaluation dataset (500 abstracts), testing dataset (500 abctracts). Each abstracts is provided by the "gold standard" that is of the keywords extracted manually by the experts. The "gold standart" has 2 subsets: "contr set" and "uncontr set". Following [6], [10], [12], [14] here we use the "uncontr set". A detailed description of the collection is given in [6].

V. Basic algorith and main problem of the research

A. Basic algorithm

All sequences (not unigrams) consisting of nouns and adjectives excluding delimiters (other grammar classes, stop words and punctuation marks) are included in the set of candidate keyphrases. The algorithm extracts all candidate keyphrases at one step. It reads the file until it finds the first noun or adjective, which is interpreted as the beginning of the phrase. The phrase is successively added by nouns and adjectives following the first word of the phrase until any of the delimiters occurs. In this case the building of the keyphrase breaks, it is added to the set of candidate keyphrases for the processed document and the scanning of the document for the next noun or adjective continues. We have used Stanford POS-tagging tool [19] and standard stop list for the algorithm.

B. Experimental background

We obtained F-score = 0.40 for the Inspec test collection considering the total output set of non-unigram candidate phrases as keyphrases. This score is higher than the previously received scores [6], [12], [14]. Noticeable, that the algorithm needs no additional data or training, extra

dictionaries, weights' assignment, etc., that is, it works independently from the dataset. We have used this basic algorithm to extract keyphrases in the test version of the system of academic search Sci-Search [15]. The shortcoming of the algorithm is that, together with "good" keyphrases it selects false keyphrases built of words belonging to the words of common scientific usage such as "experimental results", "new algorithm", "first time", that do not reveal domain-specific sense of the document. To discard such "false" keyphrases we used a manually collected list of stop-phrases including the above mentioned and similar phrases. It is obvious that manual addition of phrases to this stop-list demands considerable time expenses. In this paper we have concentrated on the ways of basic algorithm's improvement in order to move out "false" keyphrases either on the stage of candidate keyphrases' building, or postfactum. Firstly, it has been noticed that collocations of common usage consist mostly of two words. Respectively to the peculiarities of our algorithm, this means the left and the right contexts contain no adjectives or nouns, because the algorithm do not include unigrams in the set of candidate phrases. That is, if one of the tokens of the false keyphrase of 2 words enters the list of stop words, the phrase itself will not be built, because the algorithm filters the unigrams out. Some words marking the phrases of common scientific usage have been also noticed, e.g. the word "new" in "new research", "new method", "new approach", etc. If we include "new" in the list of stop words none of the three mentioned phrases will enter the set of candidate phrases. We show further in this paper that adding even a small number (about 20-100) of non-terminological words improves the quality of the keyphrases extracted by the said algorithm. Thus, the first issue the paper deals with is testing the ways of automatic stop list feeding in order to improve our algorithm. The second issue is to confirm that excluding of non-terminological candidate phrases improves the quality of keyphrases assigning and to propose a formalized alternative to manual filtering of such keyphrases.

VI. STOP LIST FEEDING

Two methods of automatic stop list renewal have been tested. The first method based only on the data about word frequencies and some speculations borrowed from the lexical statistics has been tested on the Inspec test collection and an exterior corpus consisting of 40 000 abstracts of scientific papers indexed by the Sci-Search system [15]. The second method based on the search of the stop words to improve the keyphrase extraction quality is applied to the Inspec training collection.

A. Frequency-based stop list renewal

The experiment's aim is to scan the dynamics of F-score after adding items from the frequency dictionary of the Inspec test collection the list of stop words, beginning with the high-frequency words and ending with *hapax legomena* (words having frequency = 1, unique words). The working hypothesis proposes, that lexis composing a keyphrase may enter any frequency zone, that is, it may belong to high-frequent, medium-frequent and low-frequent words [22]. The same is supposed for the items of the "false keyphrases" that should be included in the stop phrase list (e.g. "the first time", "frankly speaking", "give a notion", etc.). To exclude items of "good" keyphrases from the stop list a frequency dictionary of keyphrases is used. The tokens of the dictionary are items of automatically extracted candidate keyphrases (successions of nouns and adjectives). It means that a candidate phrase "CMOS memory logic embedded technology" shall be split into "CMOS", "memory", "logic", "embedded" and "technology". We have made an assumption that such procedure bubbles the most productive terms and term collocations belonging to the same terminological cluster to the upper part of the ranked frequency list, e.g. terminological collocations with the nucleus "technology" under 1) and words with the ranks from 1 to 30 from the frequency list of the items of candidate keyphrases in Table I (rank is a characteristic assigned to the word according its frequency: word with the maximum frequency (Fmax) gets the minimal rank.). In this paper we shall use the terms "candidate stop words" for the tokens from the Inspec test collection frequency list and "candidate non-stop words" for the tokens from the frequency list of items of the set of candidate keyphrases.

1) *CMOS memory logic embedded technology, gambling internet technology, strategy technology, mas technology, tyrol technology, key technology, acceptance technology, basic technology, web technology, navigation technology, communication technology, revolution technology, available technology*

The experiment consists of successive enlarging of the list of stop words with candidate stop words. Enlarging starts with high-frequency words and ends with the last frequency "step". Stop list growth is compensated by emerging and enlarging the list of non-stop words, which is fed from the frequency list of non-stop words. When breaking the frequency distribution into "steps" is quite clear, placing the border between high-frequency and medium-frequency items has always been disputable.

J.K.Zipf [20], [21], G.Herdan [22], [23], J.Tuldava [24] and further researchers [25], [26], [32] have shown on various text data that beginning of the rank distribution is dominantly occupied by synsemantic words and the middle part of the distribution curve contains words revealing the topic of the text. However, there is still no any satisfactory criterion telling keywords from non-keywords founded merely on statistical data for a particular text. Several criteria to distinguish synsemantic words from autosemantic ones have been proposed, among them are dynamics of the coefficient of variation [27], Hirsch *h*-point introduced to

linguistics by I.-I.Popescu and G.Altmann [28], *R*-point [29]. These criteria have been already tested on whole texts [30] and *h*-point has turned out to be the most suitable criteria.

TABLE I. WORDS WITH RANKS FROM 1 TO 31 (EXCEEDING H-POINT, ABOUT H-POINT SEE BELOW) IN THE RANKED FREQUENCY LIST OF THE ITEMS OF CANDIDATE KEYPHRASES

Word	*r*, rank	*F*, absolute frequency	Word	*r*, rank	*F*, absolute frequency
system	1	127	models	17	46
systems	2	114	performance	18	45
control	3	105	fuzzy	19	41
model	4	96	several	20	39
information	5	92	electronic	21	38
design	6	68	problem	22	35
method	7	65	management	23	34
analysis	8	64	computer	24	34
algorithm	9	63	simple	25	32
image	10	62	research	26	32
different	11	53	noise	27	32
time	12	52	methods	28	32
approach	13	51	software	29	31
linear	14	50	problems	30	31
web	15	49	technology	31	30
process	16	47			

The *h*-point is defined as the point at which the straight line between two (usually) neighboring ranked frequencies intersects the $r = f(r)$ line [28 p.24], see Fig.1:

$$h = \begin{cases} r, & \text{if } \exists\ r = f(r); \\ \dfrac{f(i)r_j - f(j)\,r_j}{r_j - r_i + f(i) - f(j)}, & \text{if } \overline{\exists}\ r = f(r). \end{cases}$$

In other words, the *h*-point is that point at which $r = f(r)$ (r – rank, $f(r)$ – absolute frequency of the token having rank r). If there is no such point, one takes, if possible, two neighboring $f(i)$ and $f(j)$ such that $f(i) > r_i$ and $f(j) < r_j$ (i, j are indexes for the neighboring frequencies and neighboring ranks).

The *h*-point has been created in scientometrics by Hirsch [31]. The h-point seems to be an important indicator in rank-frequency phenomena. As is well known, every text consists of autosemantics which bring up the theme and the concomitant information, and of synsemantics which care for correct relations between autosemantics and sentences, furnish references and modify the autosemantics. The number of synsemantics is always greater than that of autosemantics, and usually they occupy the first ranks. "The *h*-point forms a fuzzy threshold between these two kinds of words. Of course, some synsemantics seldom occurs – depending on style – and occupies some higher ranks.

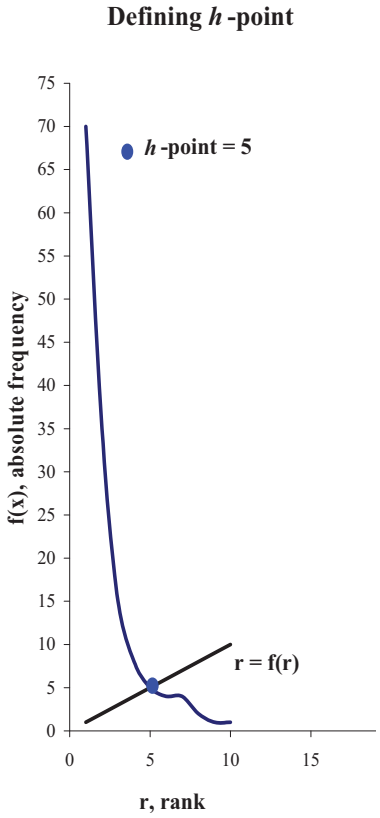

Fig.1. h-point on the distribution curve. h-point is the point where the curve of the rank distribution intersects the straight line r = f(r). r – rank, f(r) – absolute frequency

On the other hand, some autosemantics may occur more frequently than $f(h)$ and its occurrence in the pre-h domain signalizes its association to the theme of the text. In fiction one often finds proper names in the pre-h domain but in scientific and technical texts these words are always thematic words. The more autosemantics are in this domain and the more frequent they are, the greater the thematic concentration of the text" [28 p.25].

This is true in respect to the frequency list of the Inspec test collection, $h = 75$ is exceeded by the words "system", "web", "method", "model", "control", "technology" that should not be in the stop list. Words in the interval [Fmax; *h*] go alongside, besides the grammar words , with common scientific words and words of common usage like "paper", "proposed", "results", "number", "different", "presented". If the latest are added to the stop, list this can improve the quality of keyphrase extraction. Some words can not be classified into "stop" and "non-stop" ones without contextual or any additional information: e.g., "image", "approach", "performance".

To remove the words that seem suitable to be used as parts of a keyphrase from the stop list a non-stop word list is attached on the same stage. The anti-stop list is fed from

the list of candidate non-stop words to avoid the removal of domain-specific lexis. Listing 2 below shows tokens [Fmax; *h*] in the frequency list of the items of candidate keyphrases.

Tables II and III contain data about the size of the steps and the number of the words added at each step of the Inspec-based dictionaries. The results are shown in Appendix I. In Appendix I and II titles of columns show the number of words from the corpus frequency dictionary, added to the stop list – [1-350] means that words with ranks from 1 to 350 were added; titles of rows show the number of words from the frequency dictionary of items of keyphrase candidate set, added to the non-stop words list – [1-*h*] means that words with ranks from 1 to *h*-point were added. "Zero line" means we have added stop words only without balancing it with non-stop words. Splitting of the distribution curve into rank "steps" is described in Tables II-V. The obtained result (F-score = 0.41) seems to be quite satisfactory taking into account that the algorithm works with no human-aided data, additional corpora and any training.

It is necessary to comment the data in Appendix I. Firstly, the best F-score is given by adding stop words and non-stop words in equal proportions or non-stop words should prevail the stop words in 20% approximately. Even if we add full dictionaries to the stop and non-stop lists respectively we will obtain F-score = 0.41.

Secondly, there is a fact we could not give explicit notion to. When candidate non-stop words that are used 3 times are added, the F-score drops from 0.41 to 0.35 and continues to reduce with the growth of the stop-list ($1<r\leq1560$, r – rank).

The last and, probably, the most important conclusion is the profit of compensation, provided by adding words to the non-stop list. Let's consider the zero line (line "0" in Appendix I). It can be seen clearly that F-score drops from 0.41 to 0.35 and further to 0.00 without balancing the growth of the stop-list by moving valuable items out of it to the non-stop list.

Let's turn now to the outer data – the corpus of 40 000 abstracts that were indexed by the Sci-Search system which includes abstracts of the following domains: computer science, engineering, data mining and natural language processing. We applied the above mentioned procedure of stop list feeding and its' compensating with the non-stop list to the outer corpora to test whether training on exterior big data from the neighboring domain is possible. In case F-score drops, it is the symptom of inutility of training the algorithm on one corpus and forcing it to extract keyphrases from another one.

Thus, the stop list is fed from the frequency list of 40 000 abstracts and candidate non-stop words are selected from the frequency list of candidate keyphrases cut out from 40 000 abstracts. The obtained lists are applied to keyword extraction from the Inspec test collection. Tables IV and V contain data about the size of the steps and the number of the words added at each step of the exterior corpus-based dictionaries.

The results are summarized in Appendix II. There is no drastic change in F-score: 0.40 on the exterior corpus against 0.41 on the Inspec corpus. F-score decreases to 0.40 beginning with 8000 rank (absolute frequencies do not exceed 10) that can be explained by entering to the list of the words of common usage that produce "false" keyphrases in Inspec like "first time" and "rapid change". Still, one observation is worth mentioning – F-score seems frozen. If we consider zero line (line "0" in Appendix II) we shall see that the score remains unchanged even if the whole dictionary of the corpus (that is 140 000 words) is added to the list of stop words! That seems unbelievable, but it means that the autosemantic lexis of the corpora does not intersect except for the high-frequent words and the score 0.41 is gained for the initial stop list and the interval [Fmax; *h*]. This outer data can be used but the score seems unimprovable! This is yet one indirect indicator to the instability of frequencies in statistical distributions of lexis and absence of probability that is believed to be attached to each word in language. In other words, unusual dynamics of F-score points to the non-gaussian distributions in lexical statistics that is the distributions that do not held central limit theorem.

TABLE II. STOP LIST AND FEEDING. RANKS AND FREQUENCIES INVOLVED OF THE INSPEC TEST COLLECTION FREQUENCY DICTIONARY

TABLE III. COMPENSATING NON-STOP LIST FEEDING. RANKS AND FREQUENCIES INVOLVED OF THE FREQUENCY LIST OF INSPEC CANDIDATE KETPHRASES

STOPWORDS		NON-STOPWORDS	
rank, *r*	frequency,*F*	rank, *r*	frequency,*F*
1-h	3472-75	1-h	127-30
1-100	3472-55	1 –50	127-25
1-150	3472-39	1-100	127-20
1-200	3472-32	1-150	127-17
1-250	3472-28	1-200	127-14
1-300	3472-25	1-250	127-12
1-350	3472-21	1-300	127-11
1-400	3472-19	1-350	127– 9
1-500	3472-17	1-400	127– 9
1– 700	3472-12	1–450	127-8
1– 900	3472-10	1–550	127-7
1-1100	3472—8	1–650	127-6
1-1300	3472—7	1–750	127-5
1-1500	3472—6	1-1100	127-4
1-1700	3472—5	1-1560	127-3
1-2460	3472—4	1-2400	127-2
1-3270	3472—3	1-5800	127-1
1-4740	3472—2		
1-8900	3472—1		

TABLE IV. STOP LIST AND FEEDING. RANKS AND FREQUENCIES INVOLVED OF THE EXTERIOR CORPUS FREQUENCY DICTIONAR

TABLE V. COMPENSATING NON-STOP LIST FEEDING. RANKS AND FREQUENCIES INVOLVED OF THE FREQUENCY LIST OF EXTERIOR CORPUS CANDIDATE KEYPHRASES

STOPWORDS	
rank, r	frequency,F
1-h	259218-790
1-1000	259218-680
1-1500	259218-400
1-2000	259218-285
1-3000	259218-163
1-4000	259218-104
1-5000	259218– 74
1-6000	259218– 56
1-7000	259218– 44
1-8000	259218– 34
1-9000	259218– 27
1-10000	259218– 24
1-11000	259218– 20
1-12000	259218– 16
1-13000	259218– 15
1-14000	259218– 14
1-15000	259218– 13
1-16000	259218– 11
1-140000	259218– 1

h-point=790

NON-STOPWORDS	
rank, r	frequency,F
1-h	6142-411
1-750	6142-240
1-1000	6142-178
1-1500	6142-114
1-2000	6142– 80
1-3000	6142– 46
1-4000	6142– 30
1-5000	6142– 22
1-6000	6142– 16
1-7000	6142– 13
1-8000	6142– 11
1-9000	6142-9
1-10000	6142-8
1-11000	6142-7
1-12000	6142-6
1-14000	6142-5
1-17000	6142-4
1-21000	6142-3
1-29000	6142-2
1-70000	6142-1

h-point=411

B. Stop list feeding using exterior data Inspec training collection

This part of research is devoted to the task of stop list feeding using additional data. While working on the prototype of the developed system of academic search Sci-Search [15] we noticed that documents and clusters of documents are regularly annotated with the same collocations of common usage (like "experimental results"). This collocations have been collected into a separate list of "stop phrases" that was later used to remove "false" keyphrases from the set of candidate phrases. The examples of such phrases can be seen below under the letter a). These "stop phrases" have been split into a list of stop words a fragment of which can be seen below under the letter b).

a) *first experiment, future work, simple method, research project, usual problems, few method, application example e.t.c.*;

b) *high, future, research, project, several, main, usual, same, full, novel, small, new, such, result e.t.c.*

These 2 lists have been used in automatic keyword extraction from the Inspec collection (Appendix III, Appendix IV) and helped to improve the result (Appendix IV). This gives support to the hypothesis that the list of stop words stays invariant and its slight variations are domain-specific. The Inspec collection consists of the subcollections that allow to train and test the algorithms. The researchers generally use the "Test" subcollection to experiment and compare results and the "Train" subcollection to tune the parameters of the algorithm. We have assumed that building lists of stop words and stop phrases for the "Train" subcollection for their further usage for the "Test" subcollection shall enable us to improve the results of keyphrase extraction for the "Test" collection. The procedure itself includes placing each word of the "Train" collection to the list of stop words and checking the dynamics of annotating quality to "Train".

The first stage was building of the dictionary for the "Train" collection. Then each token from this dictionary was added to the list of stop words and for each word F-score was calculated (for "Train" collection). That shows how this exact token influences the quality of keyphrase extraction for the "Train" collection. If the addition of a particular word caused quality improving that exceeded a desired parameter value *a*, the word was added to the list of additional stop words. Under the quality improvement we mean the growth of F-score. When all the tokens of the dictionary have been checked, the additional list was merged with the basic stop list. Having this operation done, the algorithm was forced to analyze the "Test" collection. Results of the experiments together with parameter value *a* can be seen in Appendix V fragments of stop lists built for various values of a can be seen in Appendix VI.

Analysis of the Appendix V shows that a stop list built with the fixed a=0.0001 improves the quality of keyphrase extraction for the "Test" collection giving F-score = 0.445, that is a high result, improving the state-of-the-art, taking into consideration the fact that no additional data is used (e.g. Wikipedia). Appendix V also shows the dependence between parameter *a* and the quality of stop list feeding, that consequently influences keyphrase extraction quality for the Inspec "Test" collection. Analysis of stop words extracted with different set values of the *a* parameter shows that the parameter growing, the more stop-words are extracted, like "new" or "novel". However, the number of these words is not large, e.g. with the parameter set at a = 0.0005 only 23 of these words will be extracted. With decrease of *a* increases the total number of extracted keywords but altogether increases the number of ambivalent words, that can be stop words in the "Train" collection but are a part of keyphrase in other. This is supported by the fact that usage of a stop list built with low set value of a makes the extraction quality for the "Test" collection decrease in comparison to the usage of a stop list built with higher set value of *a*. Nevertheless, tuning the value of *a* is a trivial task for the expert because even visual control allows to decide at the very beginning of experiment whether only the words of common usage are extracted of they are mixed with domain-specific terms. The last happens, the value of *a* should be increased.

VII. CONCLUSIONS

The current paper concerns the possibilities of improving keyword extraction quality with automatic feeding of the stop words list for the basic algorithm. The

APPENDIX I. F-SCORE OBTAINED WITH AUTOMATIC FEEDING OF LISTS OF STOP-WORDS AND NON-STOP WORDS. INSPECT DATA APPLIED TO THE INSPEC DATA

		STOPWORDS																		
NON-STOP WORDS		1-h	1-100	1-150	1-200	1-250	1-300	1-350	1-400	1-500	1-700	1-900	1-1100	1-1300	1-1500	1-1700	1-2460	1-3270	1-4740	1-8900
	0	0.3502	0.3383	0.3201	0.295	0.2721	0.2618	0.2466	0.2356	0.2123	0.1777	0.1473	0.1277	0.1102	0.0986	0.084	0.5153	0.037	0.0169	0,0000
	1-h	0.4083	0.4084	0.3898	0.3646	0.3889	0.3262	0.311	0.2944	0.2727	0.2388	0.2046	0.181							0.0087
	1-50	0.4086	0.4088	0.4006	0.3814	0.3588	0.3457	0.3289	0.3167	0.2894	0.2564									0.0156
	1-100	0.4092	0.4093	0.406	0.4	0.3881	0.3762	0.3609	0.3459	0.3167	0.2825									0.0352
	1-150	0.4092	0.4093	0.4082	0.4079	0.4032	0.3968	0.389	0.3761	0.3452	0.3112									0.0568
	1-200	0.4092	0.4092	0.4092	0.4088	0.4056	0.4023	0.397	0.3891	0.3663	0.3342									0.0795
	1-250	0.4092	0.4092	0.4092	0.4096	0.4079	0.4075	0.4036	0.3994	0.3842	0.355									0.0854
	1-300	0.4092	0.4092	0.4091	0.4097	0.4096	0.4099	0.4074	0.4058	0.399	0.3783									0.1065
	1-350	0.4092	0.4092	0.4095	0.4101	0.41	0.4102	0.4094	0.4082	0.4076	0.393	0.363	0.3399							0.1221
	1-400	0.4092	0.4092	0.4095	0.4094	0.4093	0.4096	0.4088	0.4084	0.408	0.3995									0.138
	1-450	0.4088	0.4088	0.4091	0.4091	0.4089	0.4092	0.409	0.4091	0.4083	0.4023		0.3656			0.3186				0.1486
	1-550	0.4088	0.4088	0.4091	0.4092	0.4093	0.4096	0.4096	0.4097	0.4088	0.4065	0.3964	0.3851							0.1723
	1-650	0.4088	0.4088	0.4091	0.4092	0.4093	0.4094	0.4095	0.4096	0.4091	0.4065	0.4017	0.3943							0.1873
	1-750	0.4088	0.4088	0.409	0.4091	0.4091	0.4094	0.4094	0.4091	0.4088	0.408	0.4043	0.4085							0.1991
	1-1100	0.4087	0.4088	0.4088	0.4089	0.4089	0.4092	0.4092	0.409	0.4088	0.4084	0.4099	0.1276	0.405						0.2424
	1-1560	0.3503	0.3384	0.3201	0.295	0.2721	0.2618	0.2465	0.2536	0.2122	0.1776	0.1473	0.1276	0.11						0,0000
	1-2400	0.4087	0.4087	0.4087	0.4087	0.4087	0.4089	0.4088	0.4089	0.4086	0.4086	0.4096	0.4093	0.4093	0.4092	0.4092	0.4092			0.3265
	1-5800	0.4087	0.4087	0.4087	0.4087	0.4087	0.4089	0.4089	0.4089	0.4087	0.4087	0.4096	0.4096	0.4094	0.4095	0.4095	0.4095	0.4095	0.4092	0.4079

APPENDIX II. F-SCORE OBTAINED WITH AUTOMATIC FEEDING OF LISTS OF STOP-WORDS AND NON-STOP WORDS AND NON-STOP WORDS. EXTERIOR CORPUS DATA ARE APPLIED TO THE INSPEC DATA

		STOPWORDS																		
NON-STOP WORDS		1-h	1-1000	1-1500	1-2000	1-3000	1-4000	1-5000	1-6000	1-7000	1-8000	1-9000	1-10000	1-11000	1-12000	1-13000	1-14000	1-15000	1-16000	1-140000
	0	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087	0.4087
	1-h	0.4089	0.4089	0.4089	0.4089	0.4089	0.4089	0.4089	0.4089	0.4089	0.4089									0.4089
	1-750	0.4088	0.4088	0.4088	0.4088	0.4088	0.4088	0.4088	0.4088	0.4088	0.4088									0.4088
	1-1000	0.4088	0.4088	0.4088															0.4088	0.4088
	1-1500	0.4087	0.4087	0.4087																0.4087
	1-2000	0.4087			0.4087											0.4087	0.4087	0.4087		0.4087
	1-3000	0.4084		0.4084	0.4084	0.4084			0.4084						0.4084					0.4084
	1-4000	0.4062			0.4062	0.4062	0.4062							0.4063						0.4063
	1-5000	0.4063						0.4063						0.4063						0.4063
	1-6000	0.4062							0.4062				0.4063	0.4062						0.4062
	1-7000	0.4062								0.4062	0.4062									0.4062
	1-8000	0.4062								0.4062	0.4062									0.4062
	1-9000	0.4047							0.4047			0.4047								0.4047
	1-10000	0.4047						0.4047					0.4047							0.4047
	1-11000	0.4047				0.4047								0.4047						0.4047
	1-12000	0.4047													0.4047	0.4047				0.4047
	1-14000	0.4047			0.4047											0.4047	0.4047			0.4047
	1-17000	0.399																0.399		0.399
	1-21000	0.399		0.399																0.399
	1-29000	0.399																		0.399
	1-70000	0.399									0.399									0.399

APPENDIX III. THE RESULT WHEN THE LIST OF STOP PHRASES HAVE BEEN USED IN AUTOMATIC KEYWORD EXTRACTION FROM INSPEC

Collection	The number of stop-phrases	Precision	Recall	F-score
Inspec test	203	0.350	0.490	0.408

APPENDIX IV. THE RESULT WHEN THE LIST OF STOP WORDS HAVE BEEN USED IN AUTOMATIC KEYWORD EXTRACTION FROM THE INSPEC

Collection	The number of stop-words	Precision	Recall	F-score
Inspec test	266	0.37	0.47	0.418

APPENDIX V. RESULTS OF THE EXPERIMENTS WITH DIFFERENT VALUES OF PARAMETER *a*

Collection	The value of parameter *a*	The number of stop-words	Precision	Recall	F-score
Inspec test	0.0005	204	0.376	**0.499**	0.429
Inspec test	0.0002	268	0.398	**0.498**	0.442
Inspec test	0.0001	366	0.408	0.490	**0.445**
Inspec test	0.00009	398	0.408	0.488	0.444
Inspec test	0.00008	479	0.409	0.484	0.444
Inspec test	0.00007	527	0.411	0.478	0.442
Inspec test	0.00005	567	0.408	0.475	0.439
Inspec test	0.00001	3053	**0.413**	0.410	0.413

APPENDIX VI. DEPENDENCE BETWEEN PARAMETER A AND THE QUALITY OF STOP LIST FEEDING

Value of *a*	Extracted stop-words
0.0005	entire, results, various, extensions, input, main, many, number, different, way, available, large, certain,…
0.0002	basic, possible, entire, results, appropriate, controlled, actual, extensions, excellent, pure, relevant, number,
0.0001	basic, much, possible, entire, target, results, appropriate, controlled, relative, ways, latter, systematic,…
0.00007	basic, much, possible, entire, discrete-map, capacitative, target, step-and-shoot, 4-mm-diam, middle,…
0.00005	basic, much, possible, entire, discrete-map, advantages, capacitative, century, target, step-and-shoot, black,

research has shown that the feeding of the said list improves keyword extraction quality significantly: scores jump from F-score = 0.40 to F-score = 0,44 that is a high result for the algorithms used in this domain [6, 12, 14]. Alongside with the experiment based on the train collection the authors examine the ways of fully automatic stop list feeding based only on the data about word frequencies and apply the same method to Inspect test collection and exterior corpus of 40 000 abstracts indexed by the Sci-Search system. The initial point was the standard list of stop words that was, from one side, fed from the collection's frequency list and, from the other side, balanced by the list of non-stop words fed from the frequency list of the items of candidate keyphrases. The results show that using of the "alien" corpus makes F-score "frozen" but on the Inspect test collection makes the F-score rise from 0.40 to 0.41. The further work might be turned to the context features of stop words to avoid manual feeding and elaborating restrictions to the grammar structure of keyphrases to improve the quality of the candidate set.

REFERENCES

[1] Bernardini, A., Carpineto, C., *Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. Web Intelligence and Intelligent Agent Technologie*s, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on (Volume:1), 2009

[2] Zhang D. and Dong Y. Semantic, Hierarchical, Online Clustering of Web Search Results. 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, April 14-17, 2004. Proceedings, Lecture Notes in Computer Science, Springer-Verlag Berlin

[3] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, Jinwen Ma *Learning to cluster web search results. Proceeding SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 210-217

[4] Gutwina C., Paynterb G., Wittenb I., Nevill-Manningc C., Frankb E. *Improving browsing in digital libraries with keyphrase indexes. Journal Decision Support Systems - From information retrieval to knowledge management: enabling technologies and best practices archive*, Volume 27, Issue 1-2, Nov. 1999, Pages 81 – 104

[5] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proc. of IJCAI. pp. 688–673 (1999)

[6] Hulth A. *Improved automatic keyword extraction given more linguistic knowledge. Proc. of the Conference on Empirical Methods in Natural Language Processing*. – 2003. – pp. 216–223.

[7] Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. International Journal of Intelligent Systems, vol 25, pp. 1158-1186 (2010)

[8] You, W., Fontaine, D., Barhes, J.-P.: *An automatic keyphrase extraction system for scientific documents*. In: Knowl Inf Syst 34, pp. 691-724 (2013)

[9] El-Beltagy, S. R., and Rafea, A.,: KP-Miner: *A keyphrase extraction system for english and arabic documents. In: Information Systems, 34, pp. 132-144 (2009)*

[10] Tsatsaronis, G., Varlamis, I., Norvag, K.: *SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs. In: Proc. of the 23rd International Conference on Computational Linguistics*, pp. 1074–1082 (2010)

[11] Su Nam Kim, Olena Medelyan, Min-Yen *Automatic keyphrase extraction from scientific articles. Language Resources and Evaluation*, Springer 2012 Kan & Timothy Baldwin.

[12] Mihalcea R., Tarau P. TextRank: *Bringing order into texts. Proc. of the Conference on Empirical Methods in Natural Language Processing*. – 2004. – P. 404–411.

[13] Wan Xiaojun and Jianguo Xiao *Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction ACM Transactions on Information Systems*, Vol. 28, No. 2, Article 8, Publication date: May 2010

[14] Zesch T., Gurevych I. *Approximate Matching for Evaluating Keyphrase Extraction. International Conference RANLP* 2009. – Borovets, Bulgaria, 2009. – pp. 484–489.

[15] Popova, S., Khodyrev, I., Egorov, A., Logvin, S., Gulyaev, S., Karpova, M., Muromtsev, D.: Sci-Search: Academic Search and *Analysis System Based on Keyphrases. In: the 4th Conference on Knowledge engineering and semantic web, Russia. Communications in Computer and Information Science series*, V. 394 pp. 281-288, Springer (2013)

[16] Popova S. and Khodyrev I., Ranking in keyphrase extraction problem: is it useful to use statistics of words occurrences? Proceedings of Kazan University journal 2013 in press

[17] Manning, C., Raghavan, P., Schutz,e H.: Introduction to Information Retrieval. Cambridge University Press (2009)

[18] Turney, P.: Learning to Extract Keyphrases from Text. In: NRC/ERB-1057, pp. 17– 43 (1999)

[19] Standford POS tagging tool DOI: http://nlp.stanford.edu/software/tagger.shtml (09.11.2012).

[20] G.K. Zipf, Human behavior and the principle of least effort. An introduction to human ecology. Cambridge (Mass.), Addison-Wesley, 1949.

[21] Zipf G.K. *The psycho-biology of language.* Boston, 1935. — 336 p.

[22] G. Herdan, Quantitative Linguistics. Berlin, Heidelberg, London. 1964.

[23] G. Herdan, Type-token mathematics. London, 1960.

[24] Ju. Tuldava,Problemy i metody kvantitativno-sistemnogo issledovaniya leksiki = Problems and methods of quantitative researches of the lexis system. Tallinn, 1986

[25] C.Manning, H.Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, 1999.

[26] Glottometrics. No.1-25 // Eds. R.Köhler, RAM-Verlag. — 2001-2013.

[27] G.Ya. Martynenko Vvedenie v teoriju chislovoi garmonii teksta [Introduction to the Theory of Numeric Harmony of a Text]. Saint-Petersburg, 2009

[28] I-I.Popescu, J.Mačutek and G.Altmann, Aspects of word frequencies // Studies in Quantitative Linguistics. Vol.3. RAM-Verlag, 2009.

[29] B.I. Kudrin, "There are Seven Points That I Differ From Zipf" [Moi sem' otlichij ot Cipfa]. Obshhaja i prikladnaja cenologija [General and Applied coenology]. Moscow, 2007. №4. pp.25-33.

[30] L.Yu.Kovriguina*, "Methodological difference in modeling of the text statistical structure (evidence from "The Tale of The Rout of Mamai")"*, in press

[31] J.E. Hirsh, *"An index to quantify an individual's scientific research output"*, in Proc. Natl. Acad. Sci. U.S.A. 2005 November 15. — Vol.102. — № 46. P.16569–16572

[32] R.Čech, G.Altmann, *Problems in Quantitative linguistics*. Vol.2. RAM-Verlag,2011.